Module 4: Coping with Multiple Predictors

Multidimensional Splines

STAT/BIOSTAT 527, University of Washington
Emily Fox
May 9th, 2013

Nonparam. Multiple Regression

- We now consider a *d*-dimensional covariate x_i

  $x_i = (x_{i1}, \dots, x_{id})$

- In its most general form, the regression equation then takes the form

  $y = f(x_1, \dots, x_d) + \epsilon$

  or, for GLMs,

  $g(\mu) = f(x_1, \dots, x_d)$

- In principle, all of the methods we have discussed so far carry over to this case rather straightforwardly
- Unfortunately, the risk of the nonparametric estimator increases rapidly with covariate dimension *d*.

Curse of Dimensionality

- To maintain a fixed level of accuracy for a given nonparametric estimator, the sample size must increase exponentially in *d*
- Set MSE = δ

  $n \propto \left(\frac{c}{\delta}\right)^{\frac{d}{4}}$
- Why? Using data in local nbhd
  - In high dim, few points in any nbhd

  everything is far away in high dim
- Consider example with *n* uniformly distributed points in $[-1,1]^d$
  - d=1: In $[-0.1, 0.1]$, $\sim \frac{n}{10}$ obs. in interval
  - d=10 In $[-0.1, 0.1]^d$,

  roughly $n\left(\frac{0.2}{2}\right)^{10} = \frac{n}{10{,}000{,}000{,}000}$

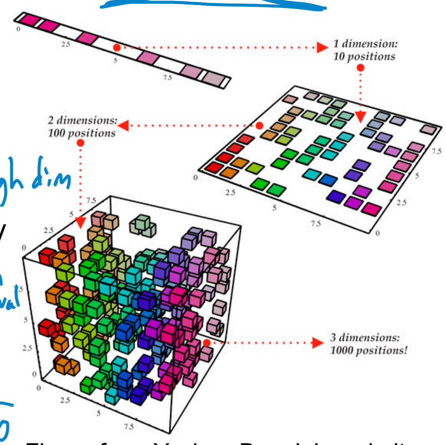


Figure from Yoshua Bengio's website

Natural Thin Plate Splines

- One-dimensional smoothing splines (obtained via regularization) can be extended to the multivariate setting as the solution to

$$\min_f \sum_{i=1}^{n} \{y_i - f(x_i)\}^2 + \lambda J(f)$$

$x_i \in \mathbb{R}^d$

- Recall roughness penalty in 1d

$$J(f) = \int f''(x)^2 dx$$

- The natural 2d extension to penalize rapid variation in either dim is

$$J(f) = \iint_{\mathbb{R}^2} \left[\left(\frac{\partial^2 f(x)}{\partial x_1^2}\right)^2 + 2\left(\frac{\partial^2 f(x)}{\partial x_1 \partial x_2}\right)^2 + \left(\frac{\partial^2 f(x)}{\partial x_2^2}\right)^2 \right] dx_1 dx_2$$

- Is the penalty affected by rotation or translation in $\mathbb{R}^2$? No!

Can extend to $d \geq 2$

Natural Thin Plate Splines

$$\min_f \sum_{i=1}^{n} \{y_i - f(x_i)\}^2 + \lambda J(f)$$ ← "bending energy"

$$J(f) = \int\int_{\mathbb{R}^2} \left[\left(\frac{\partial^2 f(x)}{\partial x_1^2} \right)^2 + 2 \left(\frac{\partial^2 f(x)}{\partial x_1 x_2} \right)^2 + \left(\frac{\partial^2 f(x)}{\partial x_2^2} \right)^2 \right] dx_1 dx_2$$

- Solution: Unique minimizer is the ***natural thin plate spline*** with knots at the x_{ij}
- Proof: See Green and Silverman (1994) and Duchon (1977)

- Similar properties and intuition as in 1d:
  - As $\lambda \to 0$, soln approaches interpolator
  - As $\lambda \to \infty$, → LS plane (no 2nd der)

Natural Thin Plate Splines

$$\min_f \sum_{i=1}^{n} \{y_i - f(x_i)\}^2 + \lambda J(f)$$

$$J(f) = \int\int_{\mathbb{R}^2} \left[\left(\frac{\partial^2 f(x)}{\partial x_1^2} \right)^2 + 2 \left(\frac{\partial^2 f(x)}{\partial x_1 x_2} \right)^2 + \left(\frac{\partial^2 f(x)}{\partial x_2^2} \right)^2 \right] dx_1 dx_2$$

- Solution: ***natural thin plate spline*** with knots at the x_{ij}
- For general λ, solution is a linear basis expansion of the form

$$f(x) = \beta_0 + \beta^T x + \sum_{j=1}^{n} b_j h_j(x)$$

with

$$h_j(x) = ||x - x_j||^2 \log ||x - x_j||$$ RBF

- Interpretation: We take an elastic flat plate that interpolates points (x_i, y_i) and penalize its "bending energy"

Natural Thin Plate Splines

$$f(x) = \beta_0 + \beta^T x + \sum_{j=1}^{n} b_j h_j(x)$$ LBE

- Coefficients are found via standard penalized LS

$$\min_{\beta,b} (y - X\beta - Eb)^T (y - X\beta - Eb) + \lambda b^T E b$$

$E_{ij} = \|x_i - x_j\|^2 \log\|x_i - x_j\|$

s.t. $\underbrace{\sum_i b_i = \sum_i b_i x_{i1} = \sum_i b_i x_{i2} = 0}_{\text{ensures finite penalty}}$

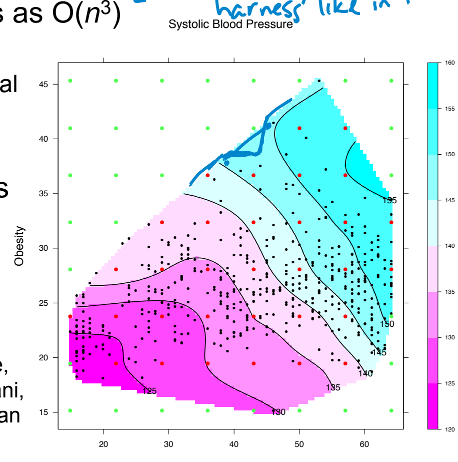
$\Rightarrow X^T b = 0$

- Interpretation: We take an elastic flat plate that interpolates points (x_i,y_i) and penalize its "bending energy"

Complexity of Thin Plate Splines

- Natural thin plate splines place knots at every location x_{ij} → lots of knots
- Computational complexity scales as O(n^3) ← no sparsity to harness like in 1d
  - Can get away with fewer knots
  - If we use K knots, then computational complexity reduces to O($nK^2 + K^3$) $K \ll n$
- Can choose some lattice of knots

heart disease data
- ignore all knots outside convex hull of data



From Hastie, Tibshirani, Friedman book

Thin Plate Regression Splines

- Thin plate regression splines truncate the "wiggly" basis b_i
- Let $E = UDU^T$ eigendecomp

  eigvec   diag matrix of eigvals
- Grab out largest k eigenvalues and eigenvectors

  D_k U_k
- Define $b = U_k b_k$
- Minimize $Eb \to U_k D_k U_k^T b$

$$\min_{\beta, b_k} (y - X\beta - U_k D_k b_k)^T (y - X\beta - U_k D_k b_k) + \lambda b_k^T D_k b_k$$

s.t. $X^T U_k b_k = 0$ (before $X^T b = 0$)

- Optimal approximation of thin plate splines using low rank basis
- Retain advantages of (i) no choice of knots, (ii) rotation invariance
- See Wood (2006) for more details

Tensor Product Splines

- Again, assume x in $\mathbb{R}^2$ (but generalizes $d \geq 2$)
- Instead of thin plate splines, consider modeling $f(x)$ as follows
- Suppose for each dimension we have a basis of functions

  $h_{1k}(x_1) \quad k = 1, \dots, M_1$

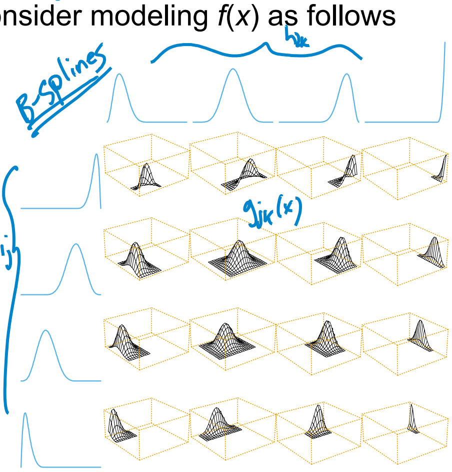
  $h_{2k}(x_2) \quad k = 1, \dots, M_2$
- Then the M_1 x M_2 dimensional ***tensor product basis*** is

$$g_{jk}(x) = h_{1j}(x_1) h_{2k}(x_2)$$

$x \in \mathbb{R}^2$

$j = 1, \dots, M_1$

$k = 1, \dots, M_2$



From Hastie, Tibshirani, Friedman book

Tensor Product Splines

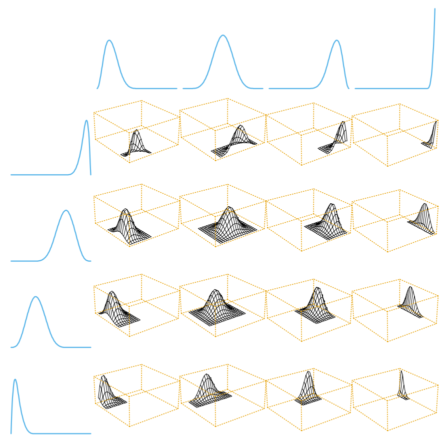
- We use this tensor product basis

$$g_{jk}(x) = h_{1j}(x_1)h_{2k}(x_2)$$

  to model *f*(*x*)

$$f(x) = \sum_{j=1}^{M_1} \sum_{k=1}^{M_2} \theta_{jk} g_{jk}(x)$$

- This formulation extends (in theory) to any dimension *d*
- Note that as the dimension of the basis grows exponentially with the input dimension *d*



From Hastie, Tibshirani, Friedman book

Tensor Product Splines Example

- Linear spline basis with L_1 truncated lines for x_1 and L_2 for x_2

$$1, x_1, (x_1 - \xi_{11})_+, \ldots, (x_1 - \xi_{1L_1})_+$$
$$1, x_2, (x_2 - \xi_{21})_+, \ldots, (x_2 - \xi_{2L_2})_+$$

- Then, the tensor product expansion is

$$f(x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$$
$$+ \sum_{\ell_1=1}^{L_1} b_{\ell_1}(x_1 - \xi_{1\ell_1})_+ + \sum_{\ell_2=1}^{L_2} b_{\ell_2}(x_2 - \xi_{2\ell_2})_+$$
$$+ \sum_{\ell_1} c_{\ell_1} x_2 (x_1 - \xi_{1\ell_1})_+ + \sum_{\ell_2} c_{\ell_2} x_1 (x_2 - \xi_{2\ell_2})_+$$
$$+ \sum\sum d_{\ell_1\ell_2}(x_1 - \xi_{1\ell_1})_+ (x_2 - \xi_{2\ell_2})_+$$

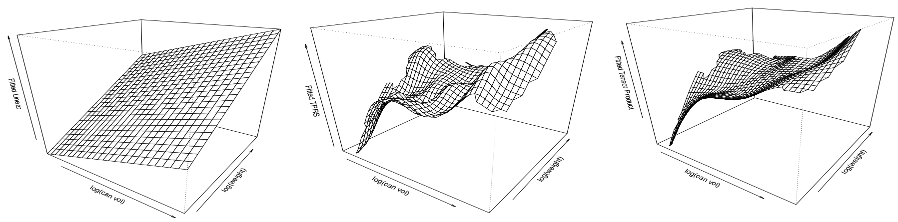
- Number of parameters:

$$4 + L_1 + L_2 + L_1 + L_2 + L_1 L_2 = (L_1+2)(L_2+2)$$

- Note: Captures interaction terms between x_1 and x_2

Tensor Product Splines Example

- For prostate cancer dataset, fits of log PSA as a function of log cancer volume and log weight for various models



Linear fit | **Thin plate regression spline** | **Tensor product spline**

pretty similar

From Wakefield textbook

Generalized Additive Models

- Both for computational reasons and added interpretability, models that assume an additive structure are very popular
- Assuming a GLM framework:

$$g(\mu(x)) = \alpha + f_1(x_1) + \dots + f_d(x_d)$$

- Is this model identifiable? No, can change α and shift f_j's to compensate $\rightarrow$ exactly same $g(\mu)$.

Fix: Constrain $\sum_{i=1}^{n} f_j(x_{ij}) = 0$

- Can model $f_j(x_j)$ using any smoother

many, many choices here (see all of module 2) or GPs...

GAM Example

- Consider using a penalized regression spline of order p_j with L_j knots for each covariate x_j

$$g(\mu) = \beta_0 + \sum_{j=1}^{d} \left[\sum_{k=1}^{p_j} \beta_{jk} x_j^k + \sum_{\ell=1}^{L_j} b_{j\ell} (x_j - \xi_{j\ell})_+^{p_j} \right]$$

- Penalization is applied to the spline coefficients b_j

$$\sum_{j=1}^{d} \lambda_j \sum_{\ell=1}^{L_j} b_{j\ell}^2$$

Comments:

- The GAM is very interpretable
  - $f_i(x_i)$ is not influenced by the other $f_j(x_j)$
  - Can plot f_j to straightforwardly see the relationship between x_i and y
- Will see that this also leads to computational efficiencies

Backfitting

- To begin, assume a standard (non-GLM) regression setting

$$y = f(x) + \epsilon$$

- For concreteness, consider

$$\min_{f_1,\ldots,f_d} \sum_{i=1}^{n} \Big(y_i - \alpha - \sum_{j=1}^{d} f_j(x_{ij}) \Big)^2 + \sum_{j=1}^{d} \lambda_j \int f_j''(t_j)^2 \, dt_j$$

- Result is an ***additive cubic spline model*** with knots at the unique values of x_{ij}
  - For X full column rank, can show that solution is unique. Otherwise, linear part of $f_j(x_j)$ is not uniquely determined
- Here, clearly $\hat{\alpha} =$ avg(y_i) $\left(\sum_i f_j(x_{ij}) = 0 \right)$
- How do we think about fitting the other parameters??

Backfitting

$y = \alpha + f_1(x_1) + \dots + f_d(x_d) + \epsilon$

- ***Backfitting*** is an iterative fitting procedure
- Since $f(x)$ is additive, if we condition on the fit of all other components $f_j(x_j)$, $j \neq i$, then we know how to fit $f_i(x_i)$

$$\underbrace{y - \alpha - \sum_{j \neq i} f_j(x_j)}_{r} = f_i(x_i) + \epsilon$$

$r \leftarrow$ partial residual ... can compute if $f_j(x_j)$ fixed

- Iterate the estimation procedure until convergence

Just like lasso, coord. ascent/descent alg.

Backfitting Algorithm

Algorithm 9.1 *The Backfitting Algorithm for Additive Models.*

1. Initialize: $\hat{\alpha} = \frac{1}{N}\sum_1^N y_i$, $\hat{f}_j \equiv 0, \forall i, j$.

init f_j

take avg., then fix

2. Cycle: $j = 1, 2, \dots, p, \dots, 1, 2, \dots, p, \dots,$

partial res.

$$\hat{f}_j \leftarrow \mathcal{S}_j \left[\{y_i - \hat{\alpha} - \sum_{k \neq j} \hat{f}_k(x_{ik})\}_1^N \right],$$

smoother chosen for x_j

fit using partial res.

numerical reasons

$$\hat{f}_j \leftarrow \hat{f}_j - \frac{1}{N}\sum_{i=1}^N \hat{f}_j(x_{ij}).$$

until the functions $\hat{f}_j$ change less than a prespecified threshold.

From Hastie, Tibshirani, Friedman book

GAMs and Logistic Regression

- A generalized additive logistic regression model has the form

$g(\mu) = \log \frac{Pr(Y=1|x)}{Pr(Y=0|x)} = \alpha + f_1(x_1) + \ldots + f_d(x_d)$

- The functions $f_1, \ldots, f_d$ can be estimated using a backfitting algorithm, too
- First, recall IRLS algorithm for *parametric* logistic regression

$$z = X\beta^{\text{old}} + W^{-1}(y - p)$$

$p = [p_1, \ldots, p_n]$

$p_i = p(x_i; \beta^{old})$

current fit

$W =$ diag matrix of $p_i(1-p_i)$

$$\beta^{\text{new}} \leftarrow \arg\min_{\beta} (z - X\beta)^T W (z - X\beta)$$

weighted LS using weights W_{ii}

backfitting w/in Newton-Raphson

GAMs and Logistic Regression

Algorithm 9.2 *Local Scoring Algorithm for the Additive Logistic Regression Model.*

1. Compute starting values: $\hat{\alpha} = \log[\bar{y}/(1-\bar{y})]$, where $\bar{y} = \text{ave}(y_i)$, the sample proportion of ones, and set $\hat{f}_j \equiv 0 \; \forall j$.

   take avg of y + fix

2. Define $\hat{\eta}_i = \hat{\alpha} + \sum_j \hat{f}_j(x_{ij})$ and $\hat{p}_i = 1/[1 + \exp(-\hat{\eta}_i)]$.

   current fit like $X^T\beta^{old}$

   current est of prob p

   Iterate:

   (a) Construct the working target variable

   $y - p$

   $$z_i = \hat{\eta}_i + \frac{(y_i - \hat{p}_i)}{\hat{p}_i(1-\hat{p}_i)}.$$

   W_{ii}^{-1}

   just like on prev. slide

   (b) Construct weights $w_i = \hat{p}_i(1 - \hat{p}_i)$

   weights

   (c) Fit an additive model to the targets z_i with weights w_i, using a weighted backfitting algorithm. This gives new estimates $\hat{\alpha}, \hat{f}_j, \forall j$

   weighted backfitting instead of weighted LS

3. Continue step 2. until the change in the functions falls below a prespecified threshold.

From Hastie, Tibshirani, Friedman book

GAM Logistic Example

- Example: *predicting spam*
- Data from UCI repository
- Response variable: *email* $\to 0$ or *spam* $\to 1$
- $k =$ 57 predictors:
  - 48 quantitative – percentage of words in email that match a give word such as “business”, “address”, “internet”,…
  - 6 quantitative – percentage of characters in the email that match a given character (; , [! $ #)
  - The average length of uninterrupted capital letters: CAPAVE
  - The length of the longest uninterrupted sequence of capital letters: CAPMAX
  - The sum of the length of uninterrupted sequences of capital letters: CAPTOT

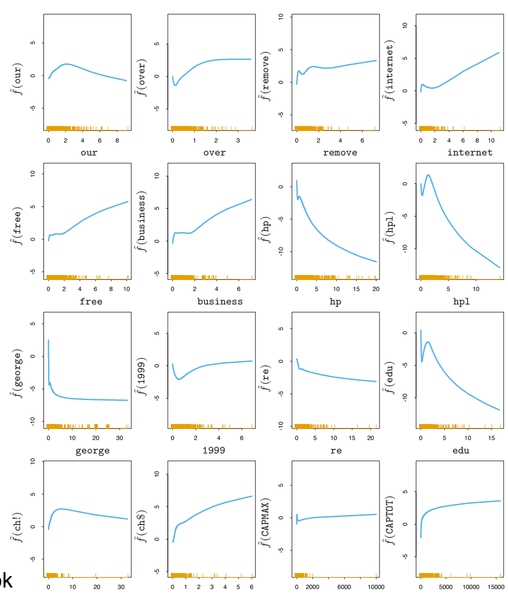
GAM Logistic Example

- Test set of 1536 emails
- Training set: n=3065
- Use a GAM with a cubic smoothing spline
  - Each with 4 dof

$tr(L_j(x_j)) = 4$

- Estimated functions for significant predictors
  - Note large discontinuity near 0 for many
- Test error of 6.6%

From Hastie, Tibshirani, Friedman book



Other GAM formulations

- Semiparametric models:

$$g(\mu) = X^T\beta + \alpha + f(z)$$

model nonparam.

model linearly

- ANOVA decompositions:

$$f(x) = \alpha + \sum_j f_j(x_j) + \sum_{j<k} f_{jk}(x_j, x_k) + \dots$$

main effects

capture interactions

Choice of:
  - Maximum order of interaction
  - Which terms to include – maybe not all main effects + interactions
  - What representation
    –reg. splines + tensor product for interaction or thin plate . . .

- Tradeoff between full model and decomposed model

Connection with Thin Plate Splines

- Recall formulation that lead to natural thin plate splines:

$$\min_f \sum_{i=1}^n \{y_i - f(x_i)\}^2 + \lambda J(f)$$

$$J(f) = \int\int_{\mathbb{R}^2} \left[\left(\frac{\partial^2 f(x)}{\partial x_1^2}\right)^2 + 2\left(\frac{\partial^2 f(x)}{\partial x_1 x_2}\right)^2 + \left(\frac{\partial^2 f(x)}{\partial x_2^2}\right)^2 \right] dx_1 dx_2$$

- There exists a $J(f)$ such that the solution has the form

- However, it is more natural to just assume this form and apply

$$J(f) = J(f_1 + f_2 + \cdots + f_d) = \sum_{j=1}^d \int f_j''(t_j)^2 dt_j$$

What you need to know

- Nothing is conceptually hard about multivariate *x*
- In practice, nonparametric methods struggle from curse of dimensionality
- Options considered:
  - Thin plate splines
  - Tensor product splines
  - Generalized additive models
  - Combinations (to model some interaction terms)

Readings

- Wakefield – 12.1-12.3
- Hastie, Tibshirani, Friedman – 5.7, 9.1
- Wasserman – 4.5, 5.12

Survey Feedback

- Lectures:
  - Useful to post reading assignments → will do!
  - Lots of material, so make clear what is expected to know → will do!

- Homeworks
  - More frequent and more in-depth
  - Less frequent/intense
  - → ???

- Recitations
  - Make same week as HW due…Was original plan and will reset to this.